**Building a custom (keras installation) NVidia container using singularity**

**Author:** Asa Thibodeau, Onur Danaci (modified it for DoubletNN)

**Objectives:**

* Learn how to set up a singularity “sandbox” from an existing image file such as docker.
* Learn how to install python packages (i.e., keras) within a sandbox
* Learn how to convert a sandbox to a “Singularity Image Format” (.sif) file
* Run example code from keras
* Run Jupyter notebooks from singularity
* Use DoubletNN notebooks

**1) Building a singularity sandbox from an existing image**

To begin, we will build a sandbox from an existing container from NVidia. Rather than pulling the docker image into a “Singularity Image Format” (.sif) file, we want to build sandbox that we can manipulate without having root access.

First, log into an interactive session and load the singularity module either by typing the command, or running:

|  |
| --- |
| bash winteractive2.sh |

|  |
| --- |
| srun --mem=4G --gres gpu:1 --pty bash  module load singularity |

Now build the sandbox directory from an NVidia docker image () using the following. You need py3-gpu-latest tags in order to get py3 and gpu support, and latest version. Details are here <https://www.tensorflow.org/install/docker>:

|  |
| --- |
| singularity build --sandbox /projects/ucar-lab/XX/tensorflow3/ docker://tensorflow/tensorflow:py3-gpu-latest |

A new directory representing the docker image will be created (tensorflow3) in your /projects/ucar-lab/XX (XX being your user name) directory:

|  |
| --- |
| [user@winter014]$ ls  tensorflow3 |

**2) Using the sandbox shell and installing additional python packages**

Now that we have built a sandbox directory containing all of the files required to run the image, we can now use shell to go into this directory (the image container) with the following command:

|  |
| --- |
| singularity shell --writable tensorflow3/ |

Building the image using a sandbox directory allows us to customize it without having root access. We can now begin installing additional python packages using pip, which will install these libraries for the python installation within the sandbox directory. In this example we will install keras:

|  |
| --- |
| pip3 install keras  pip3 install pandas  pip3 install feather  pip3 install sklearn  pip3 install matplotlib  pip3 install jupyter  pip3 install pickle |

With keras installed, the last step that we need to get our customized image running is to convert the sandbox directory to “Singularity Image Format” (.sif).

**3) Converting a sandbox directory to a SIF file**

The sandbox should now be ready to run any python code that depends on keras. If you haven’t done so already, exit the sandbox shell:

|  |
| --- |
| exit |

Now, we will build a .sif from the sandbox:

|  |
| --- |
| singularity build tensorflow3.sif tensorflow3/ |

You should now see “.sif” within your current directory:

|  |
| --- |
| [user@winter014]$ ls  Tensorflow3 tensorflow3.sif |

**3) Running example code from keras**

To test that our customized image can run keras, we will download the keras source code and run an example script. In this case, it will be the MNIST convolutional neural network example.

Download the keras source code by cloning the git repository:

|  |
| --- |
| git clone https://github.com/keras-team/keras.git |

Now run python on the mnist\_cnn.py using our custom image:

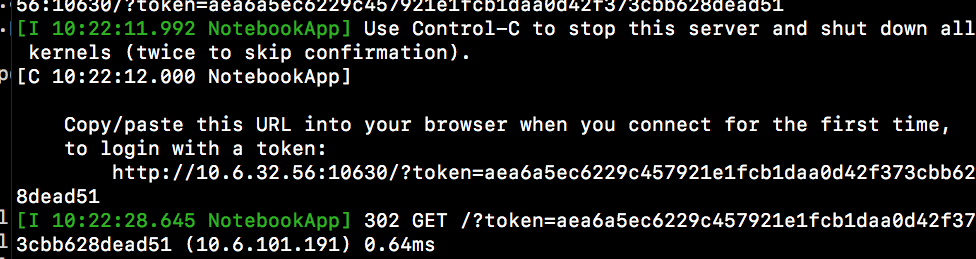
**4) Running jupyter notebooks from singularity and slurm**

Navigate to your /projects/ucar-lab/XX (XX is your username) folder again. If you type the following command, or if you run the tensorjup.sh file from that folder, you will have Jupyter notebook access

|  |
| --- |
| bash tensorjup.sh |

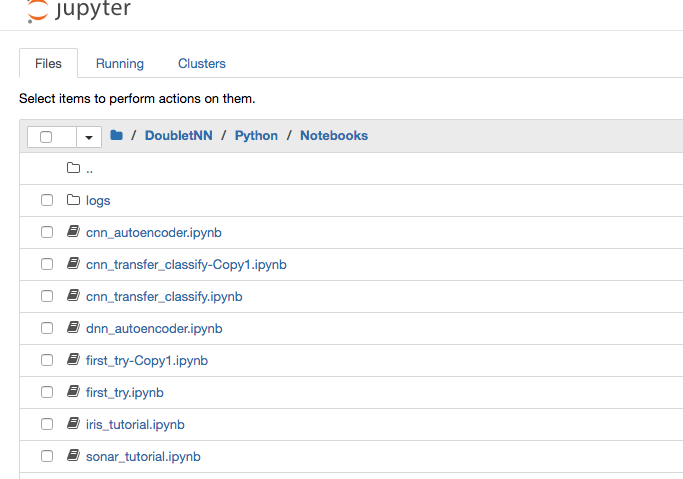
|  |
| --- |
| singularity exec --nv tensorflow3.sif jupyter notebook --no-browser --ip=${SLURMD\_NODENAME} |

This will provide you with a link (or links) on your terminal. You copy and paste this to your web browser of choice. Over there you have an interactive interface to modify files, folders, text files, scripts as well as run Jupyter notebook files





Navigate to where jupyter notebooks are. In this case the /projects/ucar-lab/XX/DoubletNN/Python/Notebooks folder:



Click one of those ipynb files. You will have notebook running. By using “shift + enter” you can evaluate the content of a given cell.

